Computer vision Seminar

수원대학교

Data Network Analysis 

데이터과학부

정인호

2023.02.09

Week4) Pytorch, GoogLeNet

1. Week 3 과제 리뷰
2. 드디어 Pytorch !
3. GoogLeNet(2014)
4. Week 4 과제설명

수원대학교
Data Network Analysis

Week4) Pytorch, GoogLeNet

1. Week 3 과제 리뷰

- 과제 피드백 – 가독성이 높은 코드

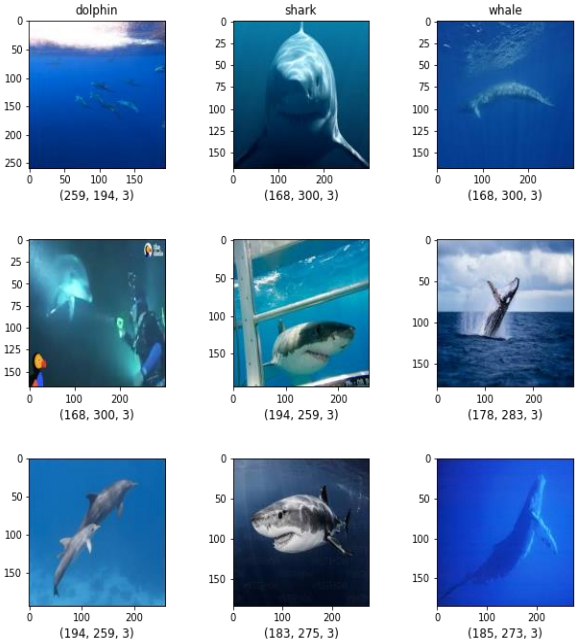
1. Week 3 과제 리뷰

과제 피드백 – 가독성이 높은 코드

[Week3 과제풀이](#)
(서채준)

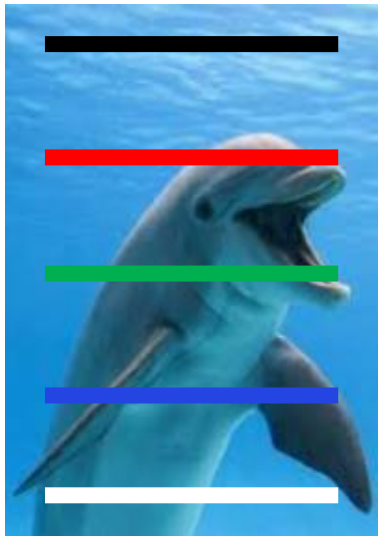
오늘은 깃허브에 같이 올려봅시다~

1번 과제 결과물



1번 결과물을 위해서는
반드시 1개의 for문 만을 이용하여
시각화 해올 것.
(2중 for문도 괜찮음)

2번 과제 결과물



1. 아무 이미지나 불러와도 된다.

2. cv2 라이브러리에서의 line을 만드는 함수를 이용하면 안되며, 반드시 imread를 한 뒤, 직접 픽셀 값을 변경하여 위 이미지를 시각화 하도록 한다.

Week4) Pytorch, GoogLeNet

2. 드디어 Pytorch !

- Pytorch 동작 방식 알아보기.
- Pytorch 데이터셋 클래스 생성하기.
- Image Augmentation

수원대학교
Data Network Analysis

2. Pytorch

Pytorch 동작 방식 알아보기.

1. 데이터 셋 준비

2. 데이터 클래스 생성

3. 모델 생성

4. 학습 시키기 !

드디어 Pytorch !
같이 코딩해봅시다.

Week4) Pytorch, GoogLeNet

3. GoogLeNet(2014)

3. ImageNet Challenge 2014

[GoogLeNet 논문]

논문 초록을 읽어보자.

구글넷 블로그 포스팅: https://inhovation97.tistory.com/45

3. ImageNet Challenge 2014

< 2. 저자들의 실험 초점 >

저자들이 연구한 내내 신경썼던 방향성이 있을 겁니다.

2. related work & **3. Motivation and High Level Considerations** 부분의 핵심 내용을 설명합니다.

(2. related work의 1x1 Conv와 다양한 사이즈의 ConV의 내용이 대략적으로 나오지만 아키텍쳐 부분에서 설명합니다.)

구글팀도 모델을 깊게 쌓기 위해서 노력을 하는데, depth와 width를 언급합니다. 아키텍쳐의 그림을 보면 아시겠지만, 여러 사이즈의 Conv를 쓰기 때문에 **width**라는 단어가 나옵니다.

모델을 깊게 쌓기위한 문제점

1. 모델을 깊게 쌓으면, 당연히 오버피팅의 문제가 존재합니다.
또한 모델이 너무 깊으면 모델의 일반화도 힘들어져서 학습셋이 적은 경우는 이미지의 디테일한 부분을 잡기가 더 더욱 힘들어집니다.

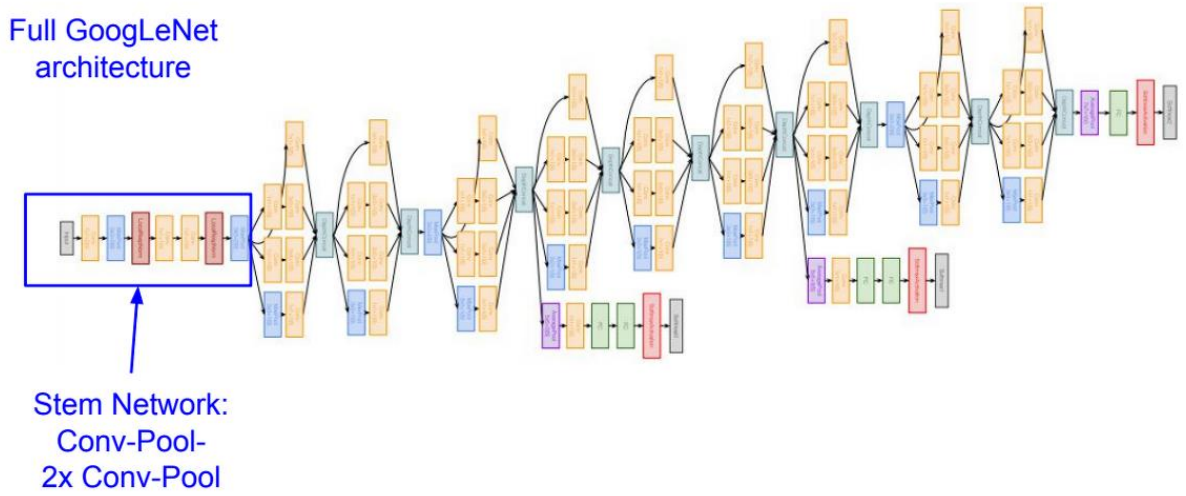
2. 네트웍의 사이즈가 깊어질수록 연산량은 훨씬 가파른 기울기로 증가합니다.
3x3의 Conv를 2개 쌓는다고 예를들면,
HxWxC가 3x3을 거쳐 HxWxC1로 나옵니다. 이걸 다시 Conv를 거치면 HxWxC2가 되는데, C1채널로 나온 피쳐맵을 또 다시 C2로 바꿔주니까 layer를 한개 더 쌓았을 뿐인데 계산량은 제곱으로 늘어나는거죠.
네트웍을 깊게 쌓되, 효율적으로 깊게 쌓아야 한다는겁니다.

3. ImageNet Challenge 2014

GoogLeNet

Case Study: GoogLeNet

[Szegedy et al., 2014]



처음 부분은 AlexNet의 구조를 그대로 가져갑니다.

vggnet과 큰 차이점은 vggnet은 처음에 큰 사이즈의 filter로 합성곱을 하는 것에 부정적이었습니다.

> vggnet의 configuration은 이제껏 우승했던 모델(AlexNet)과는 다르다고합니다.
> 첫 Conv layer를 비교적 큰 사이즈인 7x7, strid는 2를 쓴 AlexNet과는 달리 vggnet 저자들은 전체 모델에 stride는 1이며, 아주 작은 3x3필터를 쓴게 포인트라고 합니다.
> https://inhovation97.tistory.com/44

vggnet은 모델의 효율성을 위해 작은 필터만을 고집했지만, 구글넷 저자들은 관점이 달랐습니다.

input에 가까운 layer들은 굉장히 적은 범위에 correlated unit들이 분포하고 집중돼있다고 합니다.

이걸 제 생각대로 해석해보자면, 이미지 사이즈는 input일 때에 가장 사이즈가 큽니다.

그러다보니, 필터들이 그것을 전부 훑으면, 낭비되는 값들이 당연히 많을 겁니다. 그래서 AlexNet처럼 일단은 resolution을 큰 filter로 크게 줄여버리고나서 본격적인 인셉션 모듈로 들어가자는 의도였던 것 같습니다.

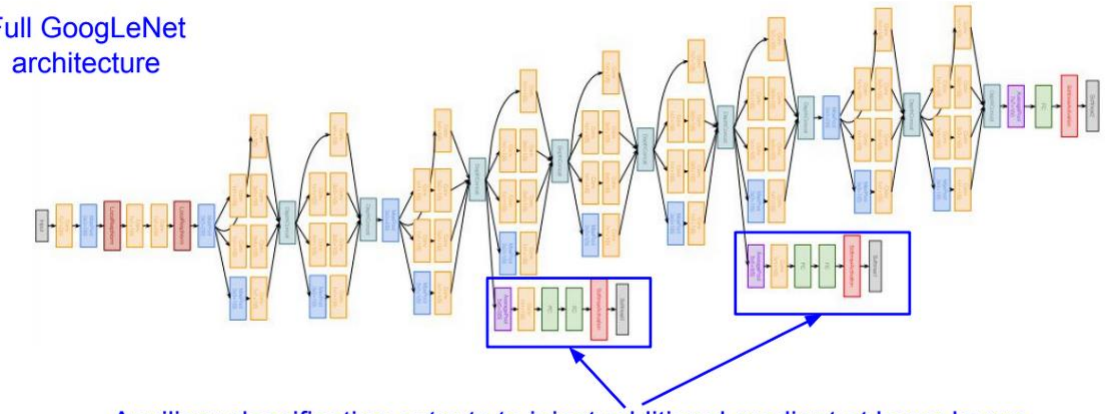
3. ImageNet Challenge 2014

GoogLeNet

Case Study: GoogLeNet

[Szegedy et al., 2014]

Full GoogLeNet architecture



Auxiliary classification outputs to inject additional gradient at lower layers (AvgPool-1x1Conv-FC-FC-Softmax)

중간 중간에 손잡이 모양의 분류기가 있습니다.

저자들은 **Auxiliary classifier**라고 하는데, 이것을 추가함으로써 backward 과정에서 깊은 네트워크의 기울기 소실 문제를 방지할 수 있습니다.

이 보조 분류기에도 똑같이 softmax 1000 class output을 설정하는데,

train시에는 Auxiliary classifier에서의 loss가 total loss에 더해지는데 0.3의 가중치를 곱해 영향력을 낮춰서 더해줍니다. 그렇기 때문에 저자들은 regularization의 효과도 기대해 볼 수 있다고 합니다. 약간의 앙상블(?) 효과도 기대해 볼 수 있는거죠.

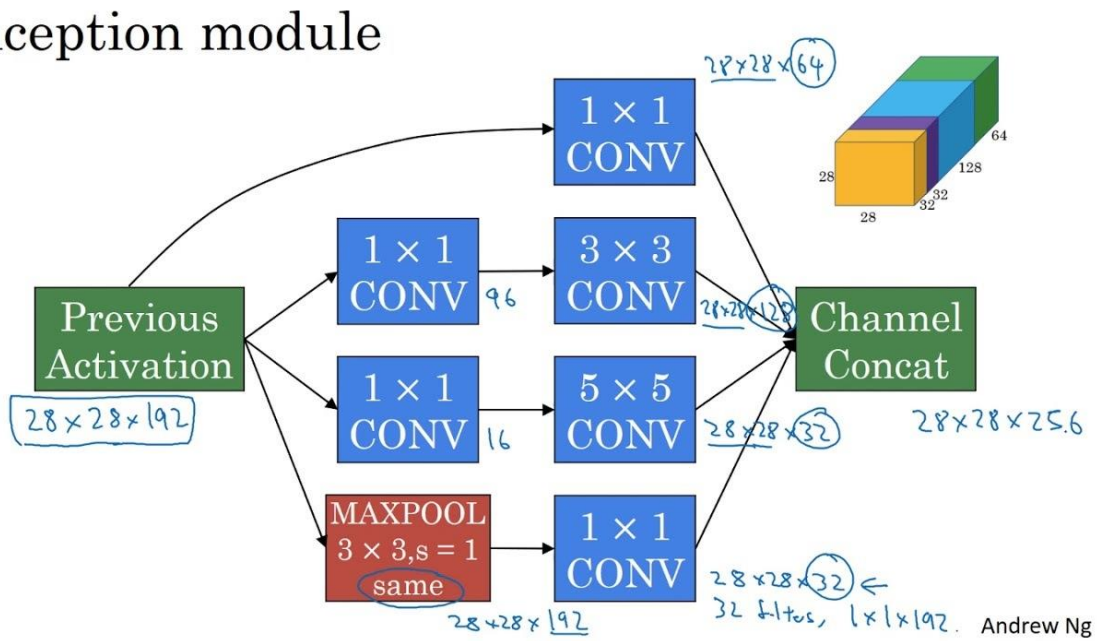
추론 시에는 쓰지 않는다고합니다.

학습때만 이용합니다.

보조 분류기, 최종 분류기를 보면 **Global Average Pooling**이 쓰입니다.

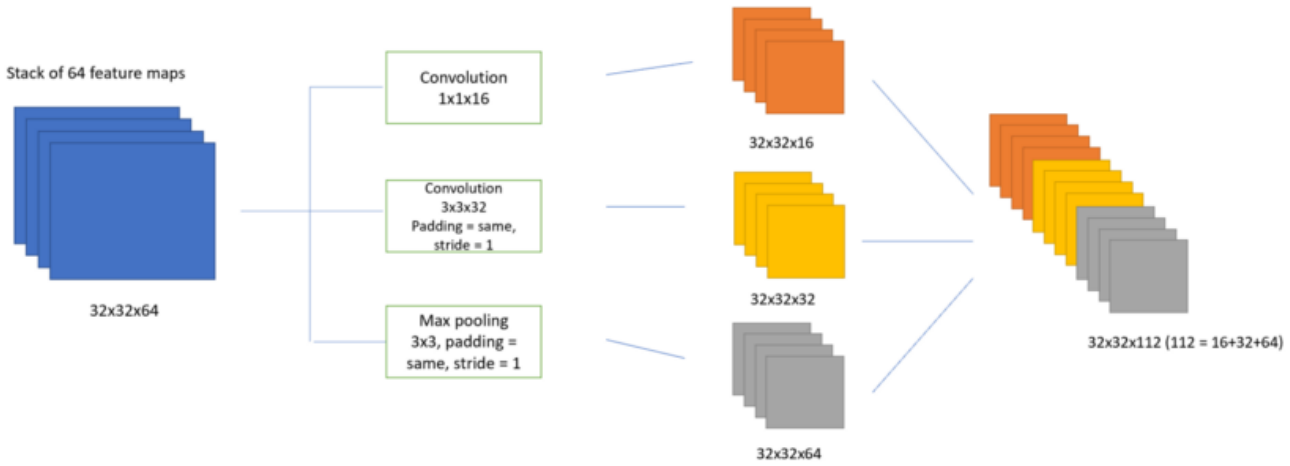
3. ImageNet Challenge 2014

CNN width를 넓힌 GoogLeNet



3. ImageNet Challenge 2014

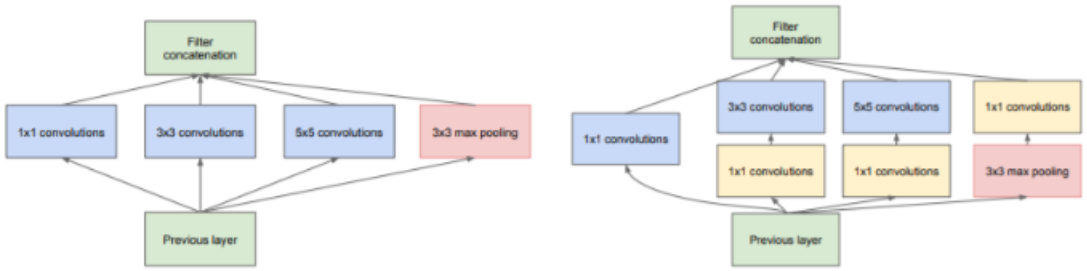
GoogLeNet



https://valentinaalto.medium.com/understanding-the-inception-module-in-googlenet-2e1b7c406106

3. ImageNet Challenge 2014

GoogLeNet



(a) Inception module, naïve version (b) Inception module with dimension reductions

Figure 2: Inception module

그렇게 인셉션 모듈을 만들고, naive version이 아닌 b를 선택하게 됩니다.

5x5 filter를 사용하면, 연산량이 너무 커지게 됩니다. 그래서 오른쪽과 같이 1x1 Conv로 channel reduction을 해줍니다.

1x1 filter가 vggnet에서도 그렇고 굉장히 중요한 역할을 한다는 것을 알 수 있습니다.

위 방식은 Cross Channel Pooling방식입니다.

maxpooling을 채널별 픽셀로 해줌으로써 같은 사이즈의 feature map을 원하는 사이즈로 down시킬 수 있는 기법입니다.

1x1 Conv는 학습할 수 있는 파라미터가 있습니다. 즉 down size도 좀 더 의미있게 할 수 있고, relu 함수도 거치기 때문에 비선형성도 한번 더 가해줄 수 있습니다. 여러 필터 사이즈로 나온 feature map들을 채널 수를 줄일 때 또 한 번 그것에 대해 고려해줄 수 있는 비선형성을 고려해줄 수 있으니 아주 좋은 방식인 거죠.

3x3 max pooling은 좀 더 다양한 resolution을 위해 추가해준 것이라고 합니다.

3x3 맥스 풀링을 할 때, 나와있지는 않지만 아마 사이즈 다운을 방지하기 위해 stride를 겹치게 할 겁니다.

그럴 경우 계속해서 특징이 강한 의미있는 픽셀의 값이 여러 번 겹치기 때문에 또 그런 부분의 장점이 있을 겁니다.

filter들의 output channel 같은 하이퍼 파라미터들은 layer depth마다 다르게 설정해줍니다.

(개인적으로는 이미 다른 발전된 모델은 보고나서 느낀 점은 이때의 구글넷이 너무 헤비해서 쓸모없는 정보도 많이 포함돼있지 않나 싶습니다. 이렇게 짜고도 비교적 간단한 vggnet과 크게 차이가 나지 않았고, backbone으로도 vggnet이 더 많이 쓰입니다.)

3. ImageNet Challenge 2014

GoogLeNet 정리하기

Week4) Pytorch, GoogLeNet

4. Week 4 과제설명

4. Week 3 과제 설명

week 4
과제 설명

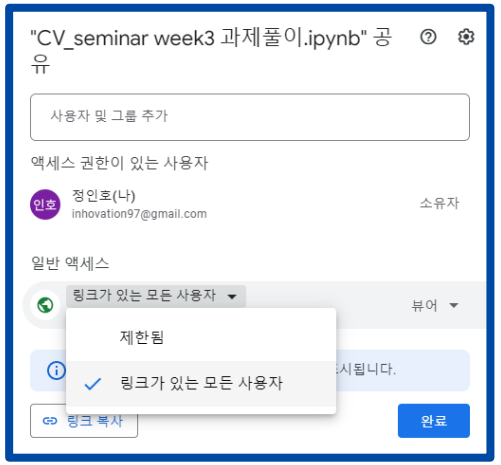
For 문을 이용하여, 이미지 시각화 해보기.
(본인의 깃헙에 올려주세요.)

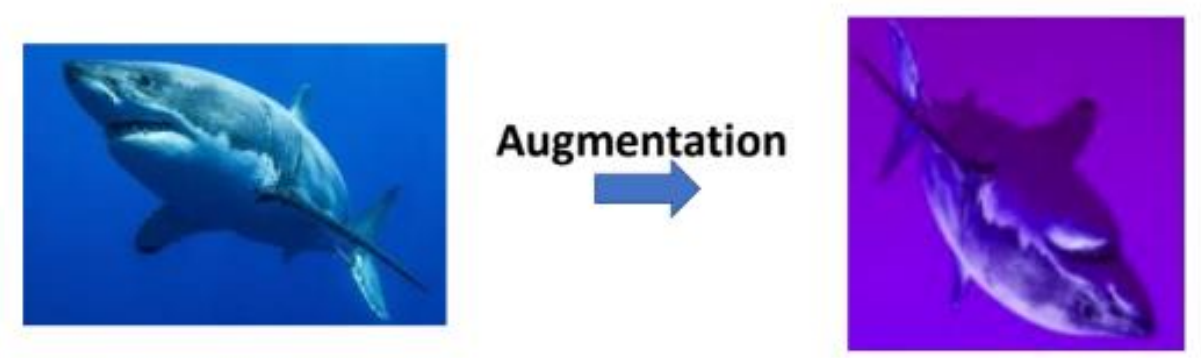
양쪽의 두 결과물을 만들어 오면 됩니다.

1. 반드시 아래의 조건을 지켜주세요 !

2. 제발 코랩 공유가능한 링크를 저에게 제출해주세요 !
(혹은 깃헙에 올린 뒤 깃헙 코드 링크를 주시면 됩니다.)

3. 기한을 지켜주세요. (다음 주 수요일 20시 까지)





1. 직접 데이터셋 클래스를 다시 짜본다.
(albumentation 툴을 이용한 augmentation 옵션까지 설정해준다.)

2. transform=None으로한 이미지와 albumentation transform을 적용한 이미지를 위처럼 시각화 해온다.

Subplot으로 나란히 시각화할 것.
Plt.imshow로 사이즈가 나오도록 시각화할 것.